REMARKS

In the Office Action dated February 7, 2006, claims 2-6, 9, 12-26, and 39-42 were rejected under 35 U.S.C. § 103 over U.S. Patent No. 6,128,640 (Kleinman) in view of U.S. Patent No. 5,598,562 (Cutler) and IBM, "Event Control Block Wait and Post Semantics for Distributed Computing Environment/Posix Threads"; and claims 37 and 38 were rejected under § 103 over Kleinman in view of Sun, "Introduction to Library Functions."

It is respectfully submitted that a *prima facie* case of obviousness has not been established with respect to independent claim 21 over Kleinman, Cutler, and IBM for at least the following reasons: (1) even if the combination of the references were proper, the hypothetical combination of Kleinman, Cutler, and IBM does not teach or suggest all elements of claim 21; and (2) no motivation or suggestion existed to combine the teachings of these references. M.P.E.P. § 2143 (8th ed., Rev. 3), at 2100-135.

The Office Action conceded that Kleinman fails to disclose the following elements of claim 21: (1) each event object having a state to indicate if the corresponding event has been signaled; (2) selectively setting a type variable to one of first and second values; (3) automatically clearing the state of the first event object to an un-signaled state and awakening only one of the plural execution entities waiting on the first event object in response to the type variable being set to the first value; and (4) not clearing the state of the first event object until manually cleared and awakening all threads waiting on the first event object in response to the type variable being set to the second value. 2/7/2006 Office Action at 10. The Office Action relied upon Cutler as disclosing an event object having a state to indicate if the corresponding event has been signaled, and on IBM as disclosing the automatically clearing and manually clearing features of claim 21. *Id.* at 11-12.

The reliance on IBM as disclosing the tasks performed at lines 10-20 of claim 21 is misplaced. Although IBM does use the words "automatically reset," "auto-reset," and "manual reset," the automatic and manual reset features of IBM are not the same as the automatically clearing and manually clearing features of claim 21. In claim 21, in response to the state of the first event object indicating that the corresponding event has been signaled, the state of the first event object is automatically cleared to an un-signaled state and *only one* of the plural execution entities waiting on the first event object is awakened in response to the type variable being set to

the first value. In contrast, in IBM, automatically resetting of an ECB (Event Control Block) is performed "when the last waiting thread is notified of the post." IBM, p. 2, ¶ 1. As further explained on page 2, in paragraph 1, of IBM, "all waiting threads" that are waiting on the ECB that has been posted are woken up, with each such waiting thread issuing a destroy call to the ECB, with the destroy calls of all threads being *ignored* until the last thread makes the destroy call, at which point a C data structure associated with the ECB is destroyed. In other words, with the automatic reset feature of IBM, all threads are awakened in response to a post of the corresponding ECB, which is contrary to the subject matter of claim 21, where *only one* of the plural execution entities waiting on the first event object is awakened in response to the state of the first event object indicating the corresponding event has been signaled and the type variable being set to the first value.

In view of the foregoing, it is clear that the hypothetical combination of Kleinman, Cutler, and IBM does not teach or suggest all elements of claim 21. The *prima facie* case of obviousness is defective for at least this reason.

Moreover, there simply did not exist any motivation or suggestion to combine the teachings of the references. Kleinman uses a notify_all function to unblock (or awaken) all threads waiting for a particular event object. Kleinman, 5:53-54, 8:37-41. Kleinman makes no suggestion whatsoever of any need to selectively awaken just one thread, or awaken all threads, based on the type of event object. Note that this teaching of Kleinman is similar to the teachings of IBM regarding awakening all threads waiting on a particular event object. There is nothing in Kleinman, Cutler, or IBM to even remotely suggest the modification of Kleinman to incorporate the feature of awakening only one of the plural execution entities waiting on the first event object in response to the type variable being set to the first value. Therefore, since no motivation or suggestion existed to combine or modify the teachings of Kleinman, Cutler, and IBM to achieve the claimed subject matter, it is respectfully submitted that the *prima facie* case of obviousness is defective for this additional reason.

Amended independent claim 9 is allowable for similar reasons.

Independent claims 37 and 38 have been cancelled, without prejudice, to render the rejection of those claims moot.

Dependent claims are allowable for at least the same reasons as corresponding independent claims.

Allowance of all claims is respectfully requested. The Commissioner is authorized to charge any additional fees and/or credit any overpayment to Deposit Account No. 14-0225 (9491).

Respectfully submitted,

Date: May 8, 2006

Dan C. Hu
Registration No. 40,025
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, TX 77024
Telephone: (713) 468-8880
Facsimile: (713) 468-8883